Performance Evaluation of Query Processing Techniques in Information Retrieval

[1]Prakasha S, [2]Shashidhar HR, [3]Dr. G T Raju, [4]Prajna Krishnan and [5]Shivaram K
[1]Research Scholar, [2]Asst. Professor, CSE dept., [3]Professor & Head, CSE dept., [4, 5]PG student
RNS Institute of Technology, Bengaluru 560098
sprakashjpg@yahoo.co.in, shashi_dhara@yahoo.com, gtraju1990@yahoo.com, prajnakrishnan@yahoo.co.in, shivaram.bhat41@gmail.com

***Abstract* - The first element of the search process is the query. The user query being on an average restricted to two or three keywords makes the query ambiguous to the search engine. Given the user query, the goal of an Information Retrieval [IR] system is to retrieve information which might be useful or relevant to the information need of the user. Hence, the query processing plays an important role in IR system.**

The query processing can be divided into four categories i.e. *query expansion, query optimization, query classification and query parsing*. In this paper an attempt is made to evaluate the performance of query processing algorithms in each of the category. The evaluation was based on dataset as specified by Forum for Information Retrieval [FIRE15]. The criteria used for evaluation are *precision* and *relative recall*. The analysis is based on the importance of each step in query processing. The experimental results show that the significance of each step in query processing and also the relevance of web semantics and spelling correction in the user query.

***Key words* - query expansion, query optimization, query parsing, web semantics and query classification**

I. Introduction

The main purpose of an IR system is to retrieve all the documents which are relevant to a user query while retrieving as few non-relevant documents as possible. The typical Information Retrieval (IR) model of the search process consists of three essentials: query, documents and search results. A user looking to fulfil information need has to formulate a query usually consisting of a small set of keywords summarizing the information need.

Queries are posed by interactive users and can be ambiguous in nature. An interactive query goes through the entire path of query parser, expansion, optimization, and processing [1].

The query processor turns user queries and data modification commands into a query plan - a sequence of operations (or algorithm) on the database from high level queries to low level commands.Complex queries are becoming commonplace, with the growing use of decision support systems. These complex queries often have a lot of common sub-expressions, either within a single query, or across multiple such queries run as a batch. The query optimization aims at exploiting common sub-expressions to reduce evaluation cost.The user queries generally suffer from low precision, or low quality document retrieval. To overcome this problem, scientists proposed methods to expand the original query with other topic-related terms extracted from exogenous (e.g. ontology, WordNet, data mining) or endogenous knowledge (i.e. extracted only from the documents contained in the collection). Methods based on endogenous knowledge, also known as relevance feedback, make use of a number of labelled documents, provided by humans (explicit) or automatic/semi-automatic strategies, to extract topic-related terms and such methods have demonstrated to obtain performance improvements [2].

A query parser, simply put, translates users search string into specific instructions for the search engine. It stands between user and the documents users are seeking, and so its role in text retrieval is vital.

The Web is rich with various sources of information. It contains the contents of documents, web directories, multimedia data, and user profile and so on. The massive and heterogeneous web document collections as well as the unpredictable querying behaviours of typical web searchers exacerbate Information Retrieval (IR) problems [13]. Hence categorization of queries allows for increased effectiveness, efficiency, and revenue potential in general-purpose web search systems. Such categorization becomes critical if the system is to return results not just from a general web collection but from topic-specific databases as well. [16]

One approach is to develop semantic Web services where by the Web services are annotated based on shared ontologies, and use these annotations for semantics-based discovery of relevant Web services. The Ontologies have been identified as the basis for semantic annotation that can be used for discovery. Ontologies are the basis for shared conceptualization of a domain, and comprise of concepts with their relationships and properties. Use of ontologies to provide underpinning for information sharing and semantic interoperability has been long realized. By mapping concepts in a Web resource (whether data or Web service) to ontological concepts, users can explicitly define the semantics of that resource in that domain. An approach for semantic Web service discovery is to have the ability to construct queries using ontological concepts in a domain. This in turn requires mapping concepts in Web service descriptions to ontological concepts. By having both the description and query explicitly declare their semantics, the results will be more relevant than keyword matching based information retrieval [10].

In this paper performance analysis is done based on the importance of each step in query processing i.e. The *query optimization*, *query expansion*, *query parsing* and *query*

DOI: 03.LSCS.2013.3.559

classification by implementing some of the algorithm from each of the mentioned categories. Experimental results shows that the significance of each steps in query processing. To evaluate the quality and effectiveness of of query processing, the two basic measures for information retrieval are used.

Precision: The fraction of retrieved documents that are relevant to the user query intent.

Recall: The fractions of relevant documents that are retrieved are in context with user query.

II. Related Work

Amit Goyal et.al analyses exponential growth in number of possible strategies with the increase in number of relations in a query has been identified as a major problem in the field of query optimization of relational databases. But as the size of a query grows, exhaustive search method itself becomes quite expensive. By modifying the A* algorithm to produce a randomized form of the algorithm and compared it with the original A* algorithm and exhaustive search [4].

Yannis E. Ioannidis primarily discuss the core problems in query optimization and their solutions, and only touch upon the wealth of results that exist beyond that. More specially, author concentrates on optimizing a single at SQL query with 'and' as the only Boolean connective in its qualification (also known as conjunctive query, select-project-join query, or non-recursive Horn clause) in a centralized relational DBMS, assuming that full knowledge of the run-time environment exists at compile time [6].

Prasan Roy et.al demonstrates that multi-query optimization using heuristics is practical, and provides significant benefits. They propose three cost-based heuristic algorithms: Volcano-SH and Volcano-RU, which are based on simple modifications to the Volcano search strategy, and a greedy heuristic [8].

Joseph M. Hellerstein defines a query cost framework that incorporates both selectivity and cost estimates for selections. The algorithm is called Predicate Migration, and proves that it produces optimal plans for queries with expensive methods [12].

Francesco Colace et.al proposes a query expansion method to improve accuracy of a text retrieval system. The technique makes use of explicit relevance feedback to expand an initial query with a structured representation called Weighted Word Pairs. Such a structure can be automatically extracted from a set of documents and uses a method for term extraction based on the probabilistic Topic Model [2].

Samer Hassan et.al describes a new approach for estimating term weights in a document, and shows how the new weighting scheme can be used to improve the accuracy of a text classifier. The method uses term co-occurrence as a measure of dependency between word features. A random-walk model is applied on a graph encoding words and co-occurrence dependencies, resulting in scores that represent a quantification of how a particular word feature contributes to a given context [5].

Mikio Yamamoto et.al describes techniques for working with much longer n-grams. Suffix arrays which were /first introduced to compute the frequency and location of a substring (n-gram) in a sequence (corpus) of length N. To compute frequencies over all $N(N + 1)/2$ substrings in a corpus, the substrings are grouped into a manageable number of equivalence classes. The paper uses these frequencies to find "interesting" substrings. Lexicographers have been interested in n-grams with high mutual information (MI) where the joint term frequency is higher than what would be expected by chance, assuming that the parts of the n-gram combine independently. Residual inverse document frequency (RIDF) compares document frequency to another model of chance where terms with a particular term frequency are distributed randomly throughout the collection. MI tends to pick out phrases with non-compositional semantics (which often violate the independence assumption) whereas RIDF tends to highlight technical terminology, names, and good keywords for information retrieval [7].

Joshua Goodman proposes two new algorithms: the *Labelled Recall Algorithm*, which maximizes the expected Labelled Recall Rate, and *the Bracketed Recall Algorithm*, which maximizes the Bracketed Recall Rate [9].

Adrian D. Thurston et.al presents two enhancements to a basic backtracking LR approach which enable the parsing of computer languages that are both context-dependent and ambiguous [11].

Kaarthik Sivashanmugam et.al develops a semantic Web services where by the Web services are annotated based on shared ontologies, and use these annotations for semantics-based discovery of relevant Web services. They also discuss one such approach that involves adding semantics to WSDL using DAML+OIL ontologies. And also uses the approach for UDDI to store these semantic annotations and search for Web services based on them [10].

Graeme Hirst et.al proposes a method for detecting and correcting many such errors by identifying tokens that are semantically unrelated to their context and are spelling variations of words that would be related to the context. Relatedness to context is determined by a measure of semantic distance [3].

III. Query Optimization

Query optimization is a function of many relational database management systems in which multiple query plans for satisfying a query are examined and a good query plan is identified. The improved A* algorithm, when used for query optimization, gives output comparable to exhaustive search in minimal amount of search space. Improved A* algorithm uses two linked lists instead of one used in original A* algorithm. It also considers global costs. Algorithm for optimizing n relations performs a large number of local optimizations. Each one starts at a random node and repeatedly accepts random downhill moves until it reaches a local minimum and it also returns the local minimum with the lowest cost found.

The three cost-based heuristic algorithms*: Volcano-SH and*

DOI: 03.LSCS.2013.3.559

Volcano-RU, which are based on simple modifications to the Volcano search strategy, and a *greedy heuristic*. The greedy heuristic incorporates novel optimizations that improve efficiency greatly.

To optimize a tree, all predicates are pushed down as far as possible, and then repeatedly apply the Series-Parallel Algorithm Using Parallel Chains to each stream in the tree, until no more progress can be made.

TABLE I. QUERY OPTIMIZATION

Optimization Algorithms	Input	Output	Precision	Recall
A* algorithm[4].	Output of original A* algorithm containing total cost of node to reach the parent node	Total Cost in global costs	17.29%	28.30%
Algorithm for optimizing n relations[8][6].	a query of N relations	• the unique set of N relations joined in the query are generated from the plans • cheapest plan/path is the final output	21.3%	41.73%
• **Volcano-SH**	directed acyclic graph of queries	decides in a cost based manner which of the nodes to materialize and share	11.85%	19.45%
• Volcano-RU	batch of queries	Optimized queries	15.36%	20.39%
• **a greedy heuristic**	Expanded DAG for the consolidated input query	Set of nodes to materialize and the corresponding. best plan	19.98%	34.38%
Predicate Migration[12]	Tree	optimally place expensive predicates in a query plan	27.37%	29.76%

From the query optimization table we infer *Algorithm for optimization n relations* performs in both the basic measures of Precision and the recall criteria of IR. The algorithm shows a precision of 21.3% and a recall of 41.73% which is comparatively higher when compared with other algorithms in the Table I.

IV. QUERY EXPANSION

In WWP feature selection the aim is to extract from a set of documents a compact representation, named Weighted Word Pairs (WWP), which contains the most discriminative word pairs to be used in the text retrieval task. In the Relations Learning stage, where graph relation weights are learned by computing probabilities between word pairs and in the Structure Learning stage, where an initial WWP graph, which contains all possible relations between aggregate roots and aggregates, is optimized by performing an iterative procedure.

In query expansion, when the algorithm *WWP feature selection-relation stage* is implemented with *the structure learning stage* algorithm there is considerable improvements in the both the precision and recall parameter of IR. It can be observed that in Table 2 that precision has almost increased by 22% and recall parameter doubled by 20%.

TABLE II: QUERY EXPANSION

Query Expansion	Input	Output	Precision	Recall
WWP feature selection- Relations Learning stage,	set of documents	vector of weighted word pairs g Discriminative word pairs to be used in the text retrieval task.	26.29%	41.63%
the Structure Learning stage	a starting WWP structure	the set of parameters _t which produces the best WWP graph	4.49%	18.90%

V. QUERY PARSING

The Labelled recall algorithm maximises expected recall rate and Bracketed recall which maximises the bracketed recall rate.

TABLE III: QUERY PARSING

Query Parsing	Input	Output	Precision	Recall
• Labelled recall algorithm[9]	Tree T_g	MAXC(1,n) contains the score of best parse, where n is length of the tree	10.94%	18.45%
• Bracketed Recall[11].	Tree T_g	Max-g. bracketed recall rate	9.80%	21.48%

In the query parsing when labelled recall algorithm is implemented with the Bracketed recall algorithm considerably improves in the recall parameter by approximately 3% even though the precision parameter remains almost the same and shown in Table III.

VI. QUERY CLASSIFICATION

The task of query classification is to assign a Web search query to one or more predefined categories, based on its topics.

A random walk uses term co-occurrence as a measure of dependency between word features. The random-walk model is applied on a graph encoding words and co-occurrence dependencies, resulting in scores that represent a quantification of how a particular word feature contributes to a given context.

Term frequency (tf) is the standard notion of frequency in corpus-based natural language processing (NLP); it counts the number of times that a type (term/word/n- gram) appears in a corpus. Document frequency (dr) counts the number of documents that contains a type at least once. Term frequency is an integer between 0 and N; document frequency is an integer between 0 and D, the number of documents in the corpus.

An algorithm based on suffix arrays is used for computing tf and dr and many functions of these quantities for all

DOI: 03.LSCS.2013.3.559

substrings in a corpus in O(NlogN) time, even though there are N(N + 1)/2 such substrings in a corpus of size N. The algorithm groups the N (N + 1)/2 substrings into at most 2N - 1 equivalence classes. By grouping substrings in this way, many of the statistics of interest can be computed over the relatively small number of classes, which is manageable, rather than over the quadratic number of substrings, which would be prohibitive.

TABLE IV: QUERY CLASSIFICATION

Query Classification	Input	Output	Precision	Recall
Random-walk model[5].	G = (V,E), the set of vertices that point to vertex Va (predecessors), and Out(Va) be the set of vertices that vertex Va points to (successors)	The algorithm terminates when the convergence point is reached for all the vertices, meaning that the error rate for each vertex falls below a pre-defined threshold.	20.19%	31.02%
• **To compute term frequency(t_f) [7].**	Suffix arrays	term frequency(tf)	26.45%	41.75%
• **To compute term document frequency (d_f)**	Suffix arrays	document frequency (dr)	13.45%	19.34%

In the query classification comparisons of various algorithms in Table 4, the computed term frequency (t_f) algorithm when implemented along with compute term document frequency (d_r) there is remarkable increase in recall and precision parameter. It is clearly shown in table 4The precision parameter increases by almost 10% and recall parameter by 20%.

VII. SPELLING ERRORS

Spelling errors that happen to result in a real word in the lexicon cannot be detected by a conventional spelling checker. A method is presented for detecting and correcting many such errors by identifying tokens that are semantically unrelated to their context and are spelling variations of words that would be related to the context. Relatedness to context is determined by a measure of semantic distance.

The algorithm for detecting and correcting malapropisms does show some improvement in recall and precision when implemented alone. But when its implemented with other algorithm it can make considerable difference in recall and precision parameters of query processing.

TABLE V: SPELLING ERRORS

Spelling errors	Input	Output	Precision	Recall
Algorithm for detecting and correcting malapropisms[3]	consideration all words in the text that	Mark a word as confirmed or unconfirmed ,raise an alarm	6.43%	8.10%

VIII. WEB SEMANTICS

The Semantic Web is a collaborative movement led by the international standards body, the World Wide Web Consortium (W3C). The standard promotes common data formats on the World Wide Web [17]. By encouraging the inclusion of semantic content in web pages, the Semantic Web aims at converting the current web dominated by unstructured and semi-structured documents into a "web of data". The Semantic Web stack builds on the W3C's Resource Description Framework (RDF).

Adding Semantics in WSDLIs to develop semantic Web services where by the Web services are annotated based on shared ontologies, and use these annotations for semantics-based discovery of relevant Web services. One such approach that involves adding semantics to WSDL is using DAML+OIL ontologies. Adding Semantics in UDDI is used to store these semantic annotations and search for Web services based on them.

Semantic Web Service Discovery, Semantic annotations added in WSDL and in UDDI are aimed at improving discovery and composition of services involves ranking based on the semantic similarity between the precondition and effect concepts of the selected operations and preconditions and effect involves ranking based on the semantic similarity between the precondition and effect concepts of the selected operations and preconditions and effect concepts of the template [10].

TABLE VI: SEMANTIC WEB TABLE

Web Semantics	Input	Output	Precision	Recall
Adding Semantics in WSDL	Message parts using XML schema constructs	Mapping Message Parts to Ontological Concepts using XML schema constructs	39.46%	51.67%
Adding Semantics in UDDI	the second tModel represent the ontologies of input	the third tModel represent the ontologies of output	29.85%	48.07%
Semantic Web Service Discovery	Phase 1: Web services (operations in different WSDL files)	matches Web services functionality provided	24.01%	39.10%
	Phase 2: result set from the first phase	Ranked first phase result set basis of semantic similarity between the input and output concepts		
	Phase 3:result set from the first phase	Ranking based on the semantic similarity between the precondition and effect concepts of the selected operations and preconditions and effect concepts of the template.		

In the web semantics from Table VI, its observed that when semantics is added to the WSDL, UDDI and Web Service Discovery there is considerable increase in the precision and recall parameters. Hence adding web semantics to the query shows a considerable improvement in performance in IR.

IX. EXPERIMENTAL EVALUATION

The experiments were conducted on the FIRE system; all the algorithms in each category of query processing, web semantics and spelling errors in query were implemented in

DOI: 03.LSCS.2013.3.559

java. A Collection of the TREC WebTrack [13] and we measured precision and recall over a set Indian statistical institute, Kolkata. Precision was measured by precision at 10 (P@10) and mean average precision [14].

Precision is the proportion of relevant documents among the retrieved documents.

Precision is defined as follows

$$\text{Precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{\{\text{retrieved documents}\}}$$

The method for estimating the recall of the algorithms is counting the number of full evaluations required to return a certain number of top results. This measure has the advantage of being independent of the software and the hardware environment, as well as the quality of the implementation.

Recall is the proportion of retrieved documents among the relevant documents.

Recall is defined as follows

$$\text{Recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{\{\text{retrieved documents}\}}$$

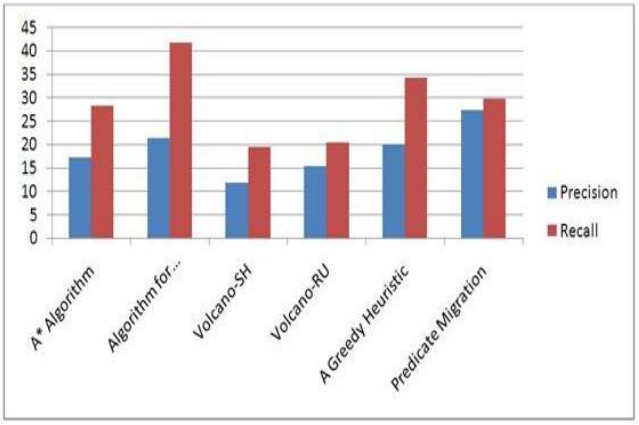

Fig 1: result of Query Optimization

In query optimization, *Algorithm for optimization n-relations* from Fig 1, provides a better results in both IR criteria precision and recall when compared to other implemented algorithms.

In query expansion from Fig 2, when WWP feature Selection-Relations Learning Stage implemented along with the Structure Learning Stage its observed that there is increases in both precision and recall parameter of IR.

In query parsing when Labelled Recall algorithm is implemented along with the Bracketed, from Fig 3, its observed that the recall parameter increases considerably.

In query classification, the Compute Term Document Frequency algorithm is implemented along with the Compute Term Frequency. From Fig 4 its observed that it improves both precision and recall parameter metrics.

In algorithm for detecting and correcting Malapropism there is seen a marginal increase in precision and recall form the Fig 5. But when implemented with the other query processing categories it can help increase in overall precision and recall parameters.

In the web semantics, from Fig 6, it's seen that by

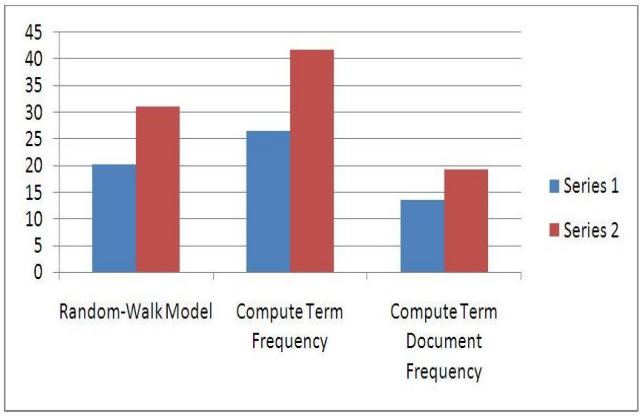

Fig 2: Results of Query Expansion

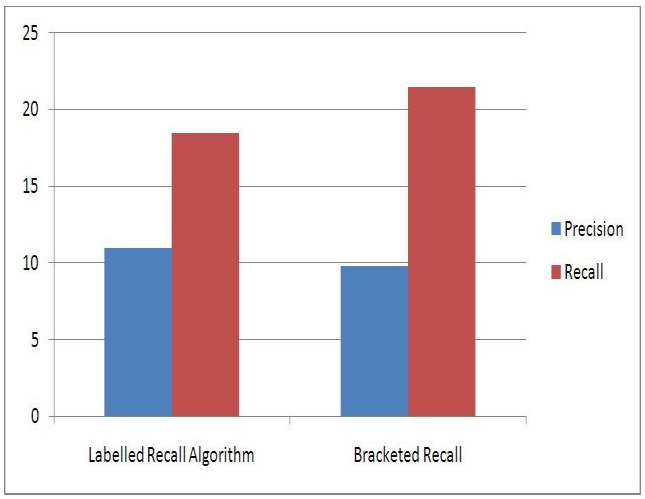

Fig 3: Results of Query Parsing

adding semantics to WSD, UDDI and Web Service Discovery shown a visible increase in precision and recall parameters.

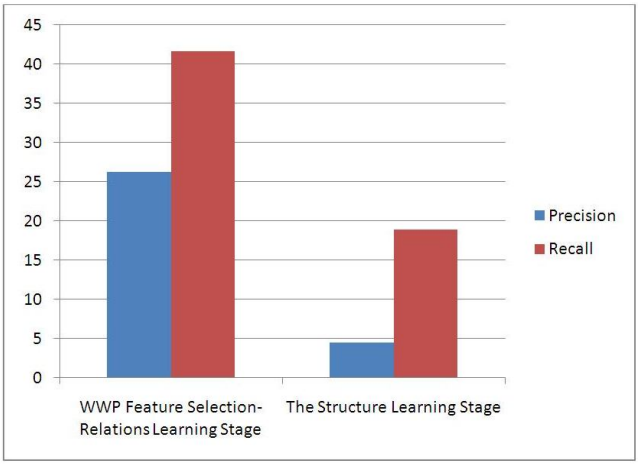

Fig 4: Results of Query Classification

For approximate query processing, web semantics and spelling error strategies (when we can have false negatives) we measure two parameters: First we measure the performance gain, as before. Second, we measure the change in recall and precision by looking at the distance between the set of original results and the set of results produced by the implemented algorithms.

DOI: 03.LSCS.2013.3.559

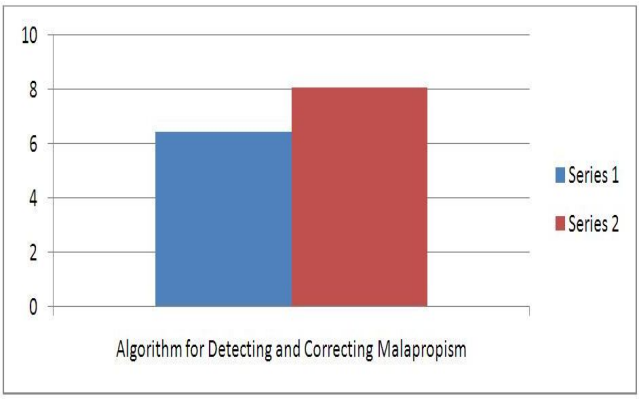

Fig 5: Results of Spelling Errors

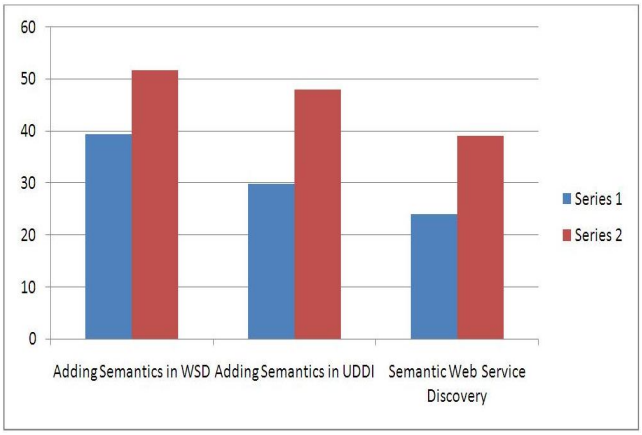

Fig 6: Results of Web Semantics

Conclusion and Future Work

The implementation of various categories of query processing demonstrates that using the query processing at various stages of IR can yield substantial gains in efficiency at no loss in precision and recall. The motivation for handling query processing is to handle the incompleteness in user query due to increasing ambiguous input from the user. In this context, a related issue is handling query imprecision– most users of online databases tend to pose imprecise queries which admit answers with varying degrees of relevance.

We have described the experiments that empirically validate our approaches in various stages of query processing. Specifically the experiments substantiate two premises.

– In order to estimate recall and precision we need to identify the set of values in a sample that represent a single real world object. In our approach these sets correspond to the result of the query processing. The experiments show that the result of the query processing may be used to improve efficiency of the search process.

– When the sample is big enough and similarity metrics are adequate for the column domain the recall/precision results are very similar to the actual recall/precision values obtained when querying the database.

However, several problems are still open. Further, the size of the samples obviously affects the results of the estimation process. In future we can work on larger dataset and also find the minimum dataset size required to estimate the precision and recall problem. More variety algorithms can be taken under the mentioned categories to get a complete picture on query processing.

References

[1] Prakasha S, H R Shashidhar, Dr. G T Raju A Survey on Various Architectures, Models and Methodologies for Information Retrieval" , IAEME, 2013.

[2] Francesco Colace et.al... "A Query Expansion Method Based On a Weighted Word Pairs Approach", Italy, National Council of Research campus, Pisa, Italy,16 – 17 January 2013

[3] Graeme Hirst et.al... "Correcting Real-Word Spelling Errors by Restoring Lexical Cohesion"

[4] Amit Goyal et.al... "Improved A* Algorithm for Query Optimization", 10th European Conference on Modelling and Simulation

[5] Samer Hassan et.al... "Random-Walk Term weighting For Improved Text Classification" by a research grant from the Texas Advanced Research Program

[6] Yannis E. Ioannidis "Query Optimization" University of Wisconsin. Partially supported by the National Science Foundation under Grants IRI-9113736 and IRI-9157368 (PYI Award) and by grants from DEC, IBM, HP, AT&T, Informix, and Oracle

[7] Mikio Yamamoto et.al. "Using Suffix Arrays to Compute Term Frequency and Document Frequency for All Substrings in a Corpus" Institute Of Information Sciences and Electronics, Tsukuba

[8] Prasan Roy et.al... "Efficient and Extensible Algorithms for Multi Query Optimization" work was supported in part by a grant from Engage Technologies/Redbrick Systems. Part of the work of Prasan Roy was supported by an IBM Fellowship

[9] Joshua Goodman "Parsing Algorithms and Metrics", Harvard university, research support from National Science Foundation Grant IRI-9350192, National Science Foundation infrastructure grant CDA 94- 01024, and a National Science Foundation Graduate Student Fellowship

[10] Kaarthik Sivashanmugam et.al... "Adding Semantics to Web Services Standards" University of Georgia.

[11] Adrian D. Thurston et.al... "A Backtracking LR Algorithm for Parsing Ambiguous Context-Dependent Languages" Queen's University supported by the Natural Sciences and Engineering Research Council of Canada

[12] Joseph M. Hellerstein "Optimization Techniques for Queries with Expensive Methods U.C. Berkeley, International Conference on Management of Data, ACM ,1997.

[13] In-Ho Kang et.al "Query Type Classification for Web Document Retrieval" SIGIR'03,July 28–August 1, 2003, Toronto, Canada, ACM

[14] "Precision and recall"http://en.wikipedia.org/wiki/Precision_and_recall, [Apr,19,2013]

[15] "Forum for Information Retrieval Evaluation (FIRE)"http://www.isical.ac.in/~clia/data.html, 4th - 6th December 2013, New Delhi [Mar,10,2012]

[16] Steven M. Beitzel et.al. "Automatic Web Query Classification Using Labelled and Unlabeled Training Data" 2005 SIGIR'05, ACM

[17] "Semantic Web" wikipedia.org/wiki/Semantic Web,[Apr, 12, 2013]

DOI: 03.LSCS.2013.3.559